



Department of Digital Business

Journal of Artificial Intelligence and Digital Business (RIGGS)

Homepage: <https://journal.ilmudata.co.id/index.php/RIGGS>

Vol. 4 No. 3 (2025) pp: 3080-3096

P-ISSN: 2963-9298, e-ISSN: 2963-914X

Pengecekan Pilihan Ganda Hasil Ujian pada Lembar Jawab secara Realtime Menggunakan Teknik Pengolahan Citra

Rizki Nur Rachmadi Yudadiningrat¹, Miftahurrahma Rosyda^{2*}

^{1,2}Program Studi Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan
miftahurrahma.rosyda@tif.uad.ac.id

Abstrak

Pendidikan memerlukan evaluasi untuk memastikan pencapaian tujuan dan efektivitas proses. Penelitian ini menangani masalah pengecekan jawaban pilihan ganda pada lembar jawaban yang tidak berbasis komputer. Metode pengolahan citra, seperti *grayscale*, *filtering*, *thresholding*, *deteksi kontur*, *warp perspective*, dan *pengurutan kontur*, digunakan untuk mendeteksi dan mengevaluasi secara otomatis lembar jawaban yang telah didigitalisasi. Metode ini menawarkan kecepatan, keandalan, dan efisiensi biaya untuk pengenalan jawaban pilihan ganda. Sistem ini, diimplementasikan menggunakan perpustakaan *OpenCV* di *Python*, mencapai akurasi sebesar 93,85% dengan jarak kamera ke lembar jawaban sekitar 17 cm. Pengujian *SUS* menunjukkan hasil kegunaan sebesar 85%, menjadikan perangkat lunak ini sebagai alat berharga untuk menilai jawaban pilihan ganda dengan efisien dan akurat dari berbagai lembar jawaban.

Kata kunci: Lembar Jawab; Pengolahan Citra; Penilaian; Pemindaian; Pilihan Ganda

1. Latar Belakang

Pendidikan, sebagai bagian penting dari perkembangan zaman modern, tak lepas dari peran guru yang berkontribusi dalam kemajuan dan pembangunan. Menurut Purwanto 1 pendidikan merupakan sebuah proses kegiatan yang disengaja atas input peserta didik untuk menimbulkan suatu hasil yang diinginkan sesuai tujuan yang ditetapkan. Muhibin Syah 2 menyatakan sebagai sebuah proses yang disengaja maka pendidikan harus dievaluasi hasilnya untuk melihat apakah hasil yang dicapai telah sesuai dengan tujuan yang diinginkan dan apakah proses yang dilakukan efektif untuk mencapai hasil yang diinginkan. Ujian digunakan sebagai standarisasi untuk menguji kelayakan, dan mengukur kemampuan serta pengetahuan seseorang atau peserta didik. Mengutip dari pernyataan Purwanto 3 ujian peserta didik dapat dipandang sebagai proses untuk merencanakan, memperoleh, dan menyediakan informasi yang sangat diperlukan dalam melakukan penilaian terhadap masing-masing peserta didik sehingga hasil evaluasi dari ujian tersebut sangatlah dibutuhkan dan apabila mungkin tersedia dengan segera.

Teknologi yang berkembang saat ini hasil dari pola pikir anak didik para guru terdahulu. Meskipun banyak teknologi dibuat untuk hiburan dan mempermudah masyarakat umum, seperti *Visual Reality* (VR), *Augmented Reality* (AR), dll, teknologi untuk mempermudah pekerjaan seorang guru, seperti *Google Docs*, *Microsoft Excel*, dll, masih terbatas. Sehingga memeriksa lembar jawaban pilihan ganda hasil ujian anak didik menjadi sangat menyulitkan bagi seorang guru.

Lembar jawaban yang tidak berbasis komputer ini tidak diciptakan sekaligus dengan sistem pengecekan yang dapat dilakukan secara otomatis. Sebuah Sekolah Menengah Pertama Negeri 1 di Wonogiri memiliki hampir sebanyak 30 s.d. 40 siswa dalam satu kelas, dengan rata-rata soal pilihan ganda suatu ujian harian berjumlah 40 s.d. 50 butir soal. Sehingga beberapa guru yang masih menggunakan jenis soal pilihan ganda dan lembar jawab yang tidak berbasis komputer akan membutuhkan lebih banyak waktu dan tenaga dalam melakukan pengecekan lembar jawabnya secara manual sebanyak anak didik yang diampunya.

Dengan perkembangan zaman, teknologi pengenalan pola dan penglihatan komputer semakin maju. Teknologi terkini yang terinspirasi dari struktur mata manusia mampu mendeteksi dan mengenali objek dalam gambar dan video serta mengolah pola menjadi informasi. Dengan teknologi ini, pengecekan lembar jawab dapat dilakukan secara cepat. *OpenCV* adalah Library Open Source untuk *Python* yang digunakan dalam *Image Processing*.

Menurut S. van der Walt 4 *OpenCV* secara teori digunakan untuk meniru cara kerja sistem penglihatan manusia dimana objek dilihat dengan “penglihatan atau mata” dan citra pada objek tersebut diteruskan dan diolah pada sebuah “sistem atau otak” dengan sedemikian rupa sehingga mengerti dan memahami objek apa yang tampak pada penglihatan tadi. Menurut Prasetyo Bobby 5 *Real Time* jika diterjemahkan ke dalam bahasa Indonesia dapat berarti sebagai waktu–nyata. Sebuah sistem *Real Time* adalah sistem yang kebenarannya secara logis didasarkan pada kebenaran hasil–hasil keluaran sistem dan ketepatan waktu pada saat hasil–hasil tersebut terjadi. Salah satu metode implementasi sistem *Real Time* adalah *Webcam* ataupun perangkat yang memiliki Hardware berupa kamera seperti Smartphone, dengan memanfaatkan fungsi video pada kamera tersebut maka segala objek yang tertangkap oleh kamera tersebut dinyatakan sebagai tertangkap secara real time atau waktu – nyata.

Maka penelitian ini akan membuat sistem yang mampu mengenali dan memeriksa pilihan ganda pada lembar jawaban yang tidak berbasis komputer secara *Real Time*. Sistem ini akan menggunakan teknik pengolahan citra untuk mengidentifikasi jawaban hasil ujian siswa.

A. Pengolahan citra

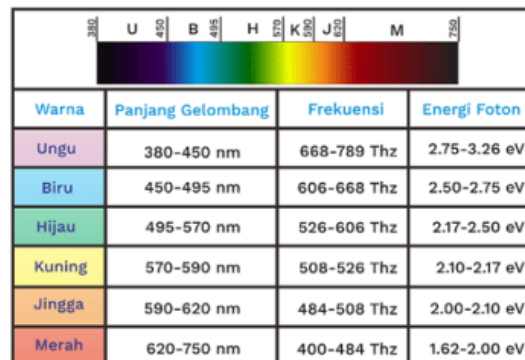
Pengolahan citra adalah proses manipulasi dan analisis citra digital untuk memperoleh informasi yang berguna. Ini mencakup konsep, metode, dan teknik untuk memahami, menganalisis, dan memanipulasi citra digital.

Beberapa konsep dasar dalam pengolahan citra meliputi representasi citra digital seperti ruang warna (RGB, HSV), resolusi, dan tingkat keabuan (Grayscale). Metode dan teknik dalam pengolahan citra mencakup tahapan seperti preprocessing, segmentasi, ekstraksi fitur, dan klasifikasi atau pengenalan pola.

OpenCV (Open Computer Vision) adalah Library Open Source untuk bahasa pemrograman Python yang mendukung teknik pengolahan citra. Dengan mengkombinasikan OpenCV dan Python dapat memberikan operasi pengolahan citra berkecepatan tinggi.

B. Ruang warna

Warna dihasilkan saat mata manusia menerima radiasi gelombang elektromagnetik atau cahaya, dengan panjang gelombang kisaran 400-700 nanometer pada 12 titik spektrum tertentu. Manusia dapat melihat dan membedakan warna karena cahaya yang dipantulkan dari objek terpapar cahaya masuk ke mata, dan otak mengasumsikan informasi tersebut sebagai warna. Misalnya, objek yang terlihat kuning memantulkan cahaya dengan panjang gelombang antara 570 hingga 590 nanometer, sehingga retina beranggapan objek itu berwarna kuning. Gambar 1 menunjukkan rentang panjang gelombang pada warna atau spektrum warna yang terlihat oleh retina.

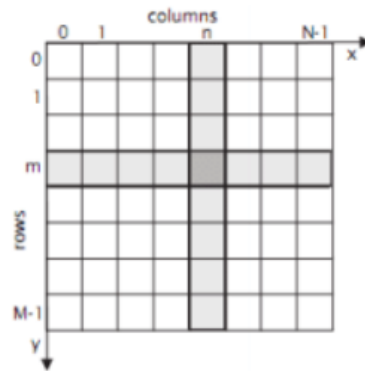


Warna	Panjang Gelombang	Frekuensi	Energi Foton
Ungu	380-450 nm	668-789 Thz	2.75-3.26 eV
Biru	450-495 nm	606-668 Thz	2.50-2.75 eV
Hijau	495-570 nm	526-606 Thz	2.17-2.50 eV
Kuning	570-590 nm	508-526 Thz	2.10-2.17 eV
Jingga	590-620 nm	484-508 Thz	2.00-2.10 eV
Merah	620-750 nm	400-484 Thz	1.62-2.00 eV

Gambar 1. Spektrum warna

C. Citra digital

Citra digital adalah representasi gambar objek nyata dalam bentuk angka yang dapat diolah oleh komputer. Citra dibagi menjadi piksel (elemen terkecil citra), dan setiap piksel direpresentasikan oleh angka yang mendeskripsikan informasi properti seperti kecerahan atau warna. Angka-angka tersebut disusun dalam susunan baris dan kolom (*Grid*) sesuai posisi vertikal dan horizontal piksel dalam gambar, seperti pada gambar 2 di bawah.



Gambar 2. Representasi citra digital dalam 2 dimensi

Menurut Mulyawan [14] proses kompresi atau pengolahan pada gambar untuk tujuan tertentu akan membutuhkan pengurangan angka pada setiap *Grid* yang disimpan secara berurutan oleh komputer.

Berdasarkan pernyataan di atas, sebuah citra dengan fungsi matriks vertikal dan horizontal dalam fungsi matematis dapat dituliskan sebagai berikut :

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (1)$$

$x \leq M-1; 0 \leq y \leq N-1; 0 \leq f(x,y) \leq G-1$

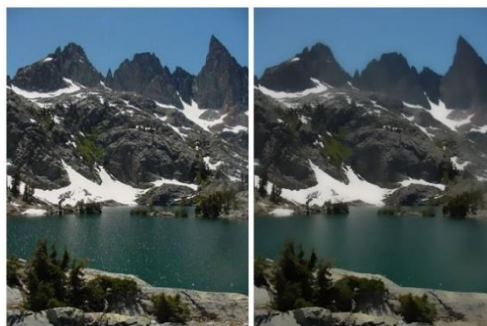
Keterangan :

- M : Jumlah baris piksel (*row*) pada matriks sebuah citra
- N : Jumlah kolom piksel (*column*) pada matriks sebuah citra
- G : Besaran skala tingkatan warna abu-abu (*gray level*)

D. Filtering

Filtering gambar adalah proses pengolahan citra untuk memperbaiki kualitas dengan menghilangkan *noise* atau derau. *Bilateral Filtering* adalah jenis *filtering* yang digunakan untuk menghilangkan noise pada gambar. *Filter bilateral* adalah teknik *non-linear* yang mengaburkan gambar tetapi mempertahankan tepian gambar yang kuat.

Menurut Sylvain Paris [15] filter bilateral adalah teknik *non-linear* yang dapat mengaburkan gambar sambil tetap mempertahankan tepian gambar yang kuat. Konsep dasarnya adalah mengurangi kontribusi piksel dengan perbedaan intensitas yang signifikan sambil mempertahankan kontribusi piksel dengan perbedaan intensitas kecil. Dengan demikian, bilateral filter dapat menghilangkan noise tanpa mengaburkan tepi atau kontur yang ada. Contoh teknik filtering dengan Bilateral Filtering dapat dilihat di gambar 3.



Gambar 3. Implementasi teknik *Bilateral Filtering*

E. Canny edge detection

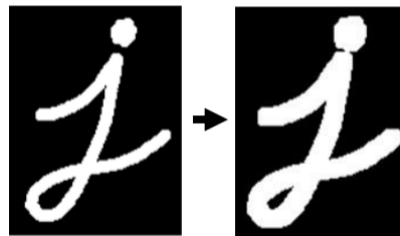
Canny Edge Detection adalah algoritma *Computer Vision* untuk mendeteksi tepi dalam citra digital. Dikembangkan oleh John F. Canny pada tahun 1986, algoritma ini sangat populer dan efektif karena menggabungkan beberapa tahap untuk menghasilkan hasil yang akurat. Metode ini digunakan untuk ekstraksi informasi struktural yang bermanfaat dalam berbagai aplikasi penglihatan komputer, mengurangi jumlah data yang akan diproses. *Canny Edge Detection* banyak diimplementasikan dalam sistem penglihatan komputer, seperti pada gambar 4 di bawah.



Gambar 4. Implementasi teknik *Canny Edge Detection*

F. Dilation

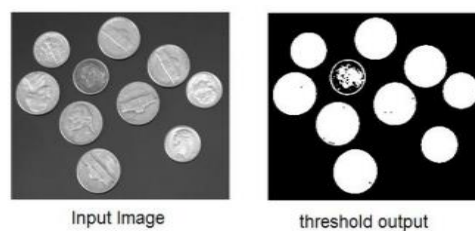
Teknik transformasi dilasi adalah salah satu cara yang efektif untuk memperbesar ukuran suatu objek pada citra. Hal ini terjadi karena ketika sebuah kernel menemukan setidaknya satu piksel bernilai 1 di bawahnya, nilai piksel tepi objek *foreground* akan berubah menjadi bernilai 1 juga. Akibatnya, daerah berwarna putih di citra akan bertambah besar dan ukuran objek di *foreground* juga akan bertambah. Contoh penerapan dilasi dapat dilihat pada gambar 5 di bawah ini.



Gambar 5. Implementasi teknik *Dilate*

G. Thresholding

Metode *Threshold* digunakan dalam segmentasi untuk memisahkan area utama dari gambar dengan latar belakangnya. Proses ini terdiri dari mengubah warna citra atau *Grayscale* ke bentuk gambar hitam putih dengan menetapkan nilai ambang batas (*Threshold*) sehingga intensitas dapat diubah menjadi 0 atau 1. Lebih jelas mengenai *Thresholding* dapat dilihat pada gambar 6 hasil implementasi *Thresholding*.



Gambar 6. Implementasi teknik *Thresholding*

H. Contours

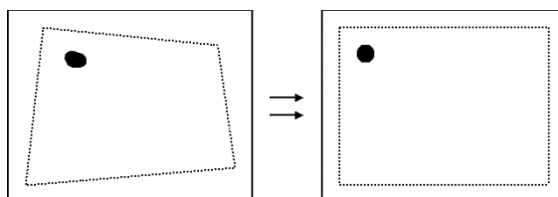
Contour adalah konsep dasar dalam *Computer Vision*, yaitu garis yang mengelilingi bagian dari gambar atau objek. *Contour* digunakan untuk mengidentifikasi bentuk, mengukur area, jarak, dan membedakan objek. Garis *contour* terdiri dari titik-titik pada citra digital atau hasil pengolahan citra. Metode *Contour* banyak digunakan dalam berbagai aplikasi *Computer Vision*, termasuk pengenalan wajah, deteksi gerak, dan pengenalan bentuk. *Contour* dapat diklasifikasikan menjadi dua jenis: *Contour* internal dan eksternal.

I. Contour sorting

Sorting merupakan fitur populer dari *Library OpenCV* yang mana banyak digunakan untuk keperluan dalam melakukan proses pengurutan, baik pada objek hasil pendeteksian maupun untuk sekedar nilai pada *array*. *Contour Sorting* dilakukan dengan cara mengurutkan kontur yang telah didapatkan baik diurutkan secara kiri ke kanan, maupun dari atas ke bawah atau sebaliknya.

J. Warp perspective

Terkadang gambar atau video untuk pemrosesan citra perlu diselaraskan agar memberikan informasi yang lebih baik. Penyelarasan ini dilakukan dengan menggunakan fungsi *Warp Perspective* di *OpenCV*. Fungsi ini memperbaiki perspektif pada citra sebelum diolah lebih lanjut. Dengan menentukan 4 titik koordinat pada citra, kita dapat menyesuaikannya dengan posisi pojok kiri atas, pojok kanan atas, pojok kiri bawah, dan pojok kanan bawah. Ilustrasi *Warp Perspective* dapat dilihat pada gambar 7.



Gambar 7. Implementasi *Warp Perspective*

K. Lembar jawaban

Lembar jawaban non-komputer adalah cara untuk menyelesaikan tugas atau tes dengan menulis jawaban di kertas. Ini telah lama digunakan di sekolah dan universitas karena biaya rendah dan mudah disebarkan. Namun, kelemahannya adalah penilaian yang lambat karena pengajar harus menilai secara manual setiap lembar jawaban, terutama jika jumlah siswa banyak atau pertanyaannya rumit. Selain itu, hasilnya mungkin tidak akurat karena kesalahan manusia dalam mencatat skor.

2. Metode Penelitian

Metode penelitian yang akan digunakan dalam kajian ini adalah berupa tindakan-tindakan yang harus dilakukan untuk menghasilkan pendeteksian jawaban otomatis pada lembar jawaban. Proses utama dalam penelitian ini terdiri dari pengumpulan dataset gambar hasil *scan* lembar jawaban, *preprocessing*, implementasi metode *canny edge detection*, implementasi metode *image thresholding*, pencarian *contour*, implementasi metode *warp perspective*, implementasi *contour sorting*, pengujian, dan penarikan kesimpulan.

A. Metode pengumpulan data

Metode pengumpulan data yang diterapkan pada penelitian ini terdiri dari 3 metode sebagai berikut :

Studi Literatur

Metode ini dilakukan dengan membaca dan menelaah beberapa jurnal dan artikel serta mempelajari data yang berkaitan dengan dataset lembar jawaban hasil pengerjaan ujian siswa.

Wawancara

Wawancara dilakukan dengan salah satu guru di sebuah Sekolah Menengah Pertama Negeri 1 Wonogiri, yaitu Ir. Yudi Iriandoyo Pati selaku guru pada mata pelajaran prakarya. Adapun hasil wawancara tersebut telah dirangkum sedemikian rupa sebagai berikut:

- Teknik penilaian yang digunakan.
- Tipe atau jenis soal yang digunakan pada saat ujian.
- Rata-rata jumlah soal setiap ujian.
- Cara melakukan penilaian manual.

Sumber Dataset

Sumber dataset adalah data lembar jawab hasil pengerjaan ujian harian siswa Sekolah Menengah Pertama yang didapatkan dari salah satu guru yang telah selesai melakukan penilaian terhadap data-data hasil pengerjaan ujian siswanya. Kemudian data lembar jawab tersebut dipilah sesuai dengan yang dibutuhkan dalam keperluan untuk melakukan pengujian.

B. Alat dan bahan

Untuk penelitian ini, kami memerlukan beberapa spesifikasi perangkat keras dan perangkat lunak. Perangkat keras yang digunakan termasuk laptop dengan spesifikasi Windows 11 Home 64-bit, GPU NVIDIA GeForce GTX 1050 Ti 4GB, RAM 16GB, dan smartphone Vivo Y91 untuk input video. Kami juga menggunakan Wifi Router dengan kecepatan koneksi 30Mbps untuk menghubungkan Droidcam. Sedangkan perangkat lunak pendukung meliputi Document Editor Microsoft Office Word 2019, browser Yandex, Python versi 3.11 (64-bit), Anaconda versi 22.9.0, dan Visual Studio Code Enterprise build 2022. Selain itu, kami menggunakan Droidcam versi 5.6.2 sebagai perangkat lunak tambahan. Untuk bahasa pemrograman, kami menggunakan Python, dan beberapa library yang digunakan adalah OpenCV versi 4.5.3, TKinter versi 8.6, Customtkinter versi 4.3, Numpy versi 1.21.2, dan SQLite3 versi 3.36.0.

Semua perangkat keras dan perangkat lunak ini sangat penting dalam menjalankan penelitian dan memproses data dengan efisien.

C. Tahap pengolahan data

Proses utama dalam penelitian ini terdiri dari pengumpulan dataset gambar hasil scan lembar jawaban, *preprocessing*, implementasi metode *contour sorting*, implementasi metode *warp perspective*, implementasi metode *image thresholding*, deteksi pilihan jawaban, pengujian, dan penarikan kesimpulan.

Adapun penjelasan langkah-langkah yang dilakukan pada setiap proses dapat dijelaskan lebih rinci sebagai berikut:

Pengumpulan dataset

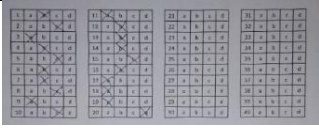
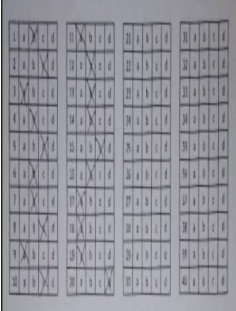
Pengumpulan dataset dalam penelitian ini, dilakukan dengan mengumpulkan lembar jawaban hasil ujian siswa yang sudah selesai dinilai dan dievaluasi dengan sampel siswa SMP N1 tahun ajaran 2021/2022 sebanyak 17 lembar jawab per ruang ujian untuk 4 kelas yang berbeda. Total dataset yang berhasil dikumpulkan adalah sebanyak 119 lembar jawaban.

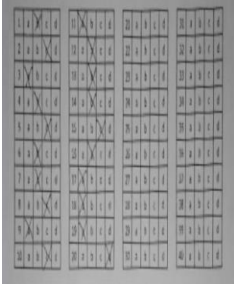
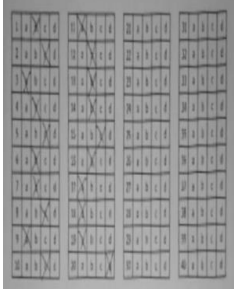
Preprocessing

Tahap pertama dalam *preprocessing* adalah input citra digital di-*resize* untuk menyamaratakan semua ukuran input citra, kemudian mengkonversi gambar berwarna menjadi gambar *grayscale*. Ini dilakukan untuk mengurangi kompleksitas gambar dan memudahkan tahapan berikutnya.

Tahap kedua adalah *filtering*, yaitu menghilangkan *noise* atau gangguan pada gambar. Hal ini bisa dilakukan dengan menggunakan filter seperti *bilateral filter* atau *gaussian blur*. Tabel 1 di bawah ini menampilkan contoh data untuk satu gambar setelah melalui tahap *preprocessing*.

Tabel 1. Alur tahap *preprocessing*

No	Tahap <i>Preprocessing</i>	Hasil
1	Data asli	
2	Mengubah ukuran gambar (resize data gambar)	

3	Menerapkan efek warna grayscale	
4	Menambahkan efek filter blur	

Canny edge detection

Pada tahap pertama dalam proses *Canny*, dataset gambar yang sudah di-*blur* di tahapan sebelumnya diambil nilai *gradient* nya. Setelah itu, *gradient* dari gambar dihitung dengan menggunakan operator Sobel untuk menemukan perubahan intensitas piksel dalam arah horizontal dan vertikal.

Selanjutnya, nilai *gradient* dihitung dan diaplikasikan pada teknik *hysteresis thresholding*, di mana nilai intensitas piksel yang lebih besar dari nilai batas atas akan dianggap sebagai tepi, sedangkan nilai yang lebih kecil dari batas bawah akan dianggap bukan tepi.



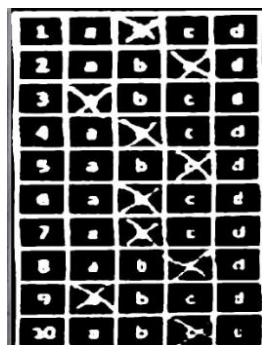
Gambar 8. Implementasi *Canny edge detection*

Penerapan dilate

Dengan melakukan dilasi setelah tahap deteksi tepi, tepi yang tadinya tipis atau tidak terhubung dengan baik akan menjadi lebih tebal dan lebih terhubung. Hal ini membantu memperbaiki dan meningkatkan representasi tepi dalam citra, sehingga mempermudah proses analisis lebih lanjut seperti segmentasi, pengenalan objek, atau pemrosesan lanjutan lainnya.

Thresholding

Dengan menggunakan teknik *thresholding*, tepian pada gambar menjadi lebih tegas dan dapat dipisahkan dengan latar belakang.



Gambar 9. Implementasi metode *Thresholding*

Hasil dari tahap *thresholding* ini adalah gambar dengan tepian yang lebih jelas dan tegas terlebih setelah dilakukan dilasi sehingga memudahkan dalam proses analisis dan pengolahan lebih lanjut, sebagaimana yang terlihat pada gambar 9 di atas.

Implementasi metode contour

Langkah selanjutnya adalah menggunakan metode *contour* untuk mendeteksi dan mengidentifikasi bentuk-bentuk pada gambar hasil *thresholding*. Pada langkah ini, dilakukan pencarian *contour* dengan beberapa kriteria, diantaranya adalah : memiliki 4 titik koordinat, memiliki luas area *contour* lebih dari 5000. Fungsi dari tahap ini adalah untuk mendapatkan tabel jawaban pilihan ganda pada lembar jawaban.

Implementasi warp perspective

Tahap selanjutnya adalah menggunakan metode *warp perspective* untuk mengubah perspektif gambar berdasarkan kontur yang telah terdeteksi. Metode ini membuat objek yang ada dalam gambar terlihat datar atau sejajar dengan bidang gambar.

Program mengidentifikasi kontur yang mewakili objek yang ingin ditransformasikan perspektifnya, dan memilih 4 titik sudut dari kontur tersebut. Selanjutnya program akan menghitung matriks transformasi perspektif yang diperlukan berdasarkan titik-titik sudut tersebut.

Dengan matriks transformasi perspektif yang telah didapatkan, program menerapkannya pada gambar asli. Hasil transformasi perspektif menampilkan objek yang semula miring atau terdistorsi menjadi terlihat datar atau sejajar dengan bidang gambar. Seperti pada gambar 10 di bawah ini.



Gambar 10. Implementasi *warp perspective*, sebelum dan sesudah

Deteksi jawaban dengan contour sorting

Metode *contour sorting* dilakukan untuk mengurutkan kontur-kontur pada citra hasil perspektif warp berdasarkan luas area. Kontur-kontur tersebut diurutkan secara menurun berdasarkan area yang mereka tutupi. Prosesnya dimulai dengan menghasilkan sejumlah kontur yang mewakili objek-objek dalam gambar. Kemudian, empat kontur terbesar yang paling signifikan dipilih dengan fungsi ``sorted()`` berdasarkan luas area menggunakan parameter ``cv2.contourArea``. Dalam kode yang diberikan, hanya kontur-kontur yang terletak pada kotak pilihan ganda yang diambil dengan *slicing*, ditentukan oleh hasil mod dari total kotak pilihan ganda. Dengan metode ini, peneliti dapat mengambil kontur-kontur yang paling relevan dan signifikan dalam gambar, sambil mengabaikan detail yang kurang penting.

3. Hasil dan Diskusi

Terdapat beberapa tahapan sebelum mulai masuk ke tahapan deteksi jawaban dengan *contour sorting*, Berikut di bawah ini adalah pembagian tahapan yang diperlukan untuk melakukan pengecekan jawaban pada citra lembar jawab digital dengan *contour sorting* :

A. Membagi citra menjadi sejumlah kotak kecil

Pembagian citra menjadi kotak-kotak kecil diperlukan guna mendapatkan jawaban siswa dan melakukan pengecekan jawaban. Proses ini digunakan untuk membagi citra hasil *warp perspective* menjadi kotak-kotak kecil berdasarkan jumlah kotak pertanyaan dan pilihan jawaban yang ditambahkan dengan 1, sehingga memisahkan setiap pilihan jawaban dalam bentuk kotak-kotak terpisah yang siap untuk diproses lebih lanjut.



Gambar 11. Pembagian kolom pilihan ganda menjadi kotak-kotak kecil

B. Mendapatkan jawaban siswa

Untuk mendapatkan kolom pilihan ganda di mana siswa memilih jawaban, diperlukan perhitungan jumlah piksel yang tidak nol (*non-zero pixel*) yang terbanyak untuk setiap kolomnya. Proses ini dilakukan setelah pemisahan kolom menjadi kotak-kotak kecil sebelumnya. Hasil perhitungan *non-zero pixel* akan disimpan dalam sebuah list baru yang akan dijadikan sebagai pilihan jawaban dari siswa.

C. Mengecek jawaban siswa

Setelah memperoleh matriks dengan nilai piksel yang bukan nol, langkah berikutnya adalah membandingkannya dengan array kunci jawaban. Dalam tahap pengembangan, *array* jawaban atau variabel *self.ans* telah ditetapkan dengan nilai 2, 3, 1, 2, 3, 2, 2, 3, 1, 3, 1, 2, 2, 2, 3, 2, 1, 1, 1, 4 seperti yang tertera pada tabel 2 di bawah ini.

Tabel 2. Nilai default variabel

No	Nama Variabel	Penjelasan	Nilai Default
1	<i>queperbo</i> <i>x (int)</i>	Jumlah nomor per tabel	10
2	<i>question</i> <i>(int)</i>	Jumlah seluruh pertanyaan	20
3	<i>choice</i> <i>(int)</i>	Jumlah banyak pilihan / opsi (a, b, c, ..)	4
4	<i>ansid</i> <i>(int)</i>	Menyimpan <i>index</i> ke- untuk jawaban	0
5	<i>ans (2D</i> <i>array,</i> <i>int)</i>	Menyimpan jawaban untuk dijadikan perbandingan pada tahap pengecekan jawaban siswa	2, 3, 1, 2, 3, 2, 2, 3, 1, 3, 1, 2, 2, 2, 3, 2, 1, 1, 1, 4

Dalam *array* kunci jawaban tersebut, nilai 1 mengindikasikan pilihan a, nilai 2 mengindikasikan pilihan b, dan seterusnya. Visualisasi hasil perbandingan *jawabanIndex* dengan jawaban yang sebenarnya dapat terlihat pada gambar 12 berikut ini.

1	a	b	c	d	14	a	b	c	d	27	a	b	c	d
2	a	b	c	d	15	a	b	c	d	28	a	b	c	d
3	a	b	c	d	16	a	b	c	d	29	a	b	c	d
4	a	b	c	d	17	a	b	c	d	30	a	b	c	d
5	a	b	c	d	18	a	b	c	d	31	a	b	c	d
6	a	b	c	d	19	a	b	c	d	32	a	b	c	d
7	a	b	c	d	20	a	b	c	d	33	a	b	c	d
8	a	b	c	d	21	a	b	c	d	34	a	b	c	d
9	a	b	c	d	22	a	b	c	d	35	a	b	c	d
10	a	b	c	d	23	a	b	c	d	36	a	b	c	d
11	a	b	c	d	24	a	b	c	d	37	a	b	c	d
12	a	b	c	d	25	a	b	c	d	38	a	b	c	d
13	a	b	c	d	26	a	b	c	d	39	a	b	c	d

Gambar 12. Visualisasi perbandingan jawaban siswa dengan kunci jawaban

D. Pengujian system

Terdapat empat pengujian pada penelitian ini, berikut adalah diantaranya :

Pengujian kesesuaian posisi tabel jawaban

Pada pengujian ini dilakukan pengecekan kesesuaian posisi tabel jawaban pilihan ganda pada suatu lembar jawaban, pengujian ini dapat dilakukan jika terdapat dua atau lebih tabel pada lembar jawaban yang dihadapkan di depan kamera dan berhasil diidentifikasi setelah melewati beberapa tahap pengecekan.

Pengujian ini sangat bergantung pada dua hal, yaitu tingkat kecerahan dan jarak, jika lembar jawaban terlalu gelap maka terdapat kemungkinan tabel tidak terdeteksi, dan jika terlalu jauh maka akan menyebabkan *contour* tabel tidak tersambung dan gagal melewati tahap pengecekan pertama.

Untuk mendapatkan akurasi dari pengujian ini, dapat menggunakan rumus *confusion matrix* sebagai berikut :

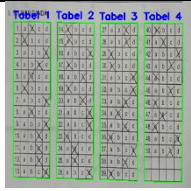
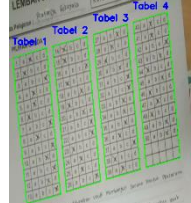
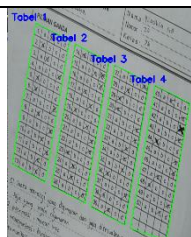
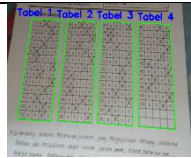
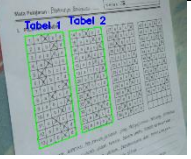
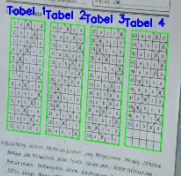
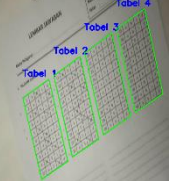
$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2)$$


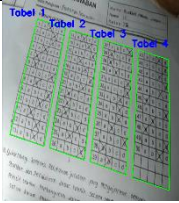

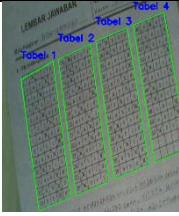
Keterangan :

- TP = (*True Positif*) Saat posisi dan jumlah tabel sesuai dengan posisi dan jumlah tabel pada dataset lembar jawaban.
- FP = (*False Positif*) Saat posisi dan jumlah tabel diidentifikasi tetapi tidak sesuai dengan posisi atau jumlah tabel pada dataset lembar jawaban.
- TN = (*True Negatif*) Saat tidak ada tabel yang diidentifikasi dan kondisi menyatakan posisi atau jumlah tabel tidak sesuai.
- FN = (*False Negatif*) Saat tidak ada tabel yang diidentifikasi tetapi kondisi menyatakan posisi atau jumlah tabel sesuai dengan dataset.

Tabel di bawah ini merupakan hasil dari pengujian kesesuaian tabel pilihan ganda yang diambil dari berbagai macam sudut atau *perspective*.

Tabel 3. Hasil pengujian kesesuaian tabel pilihan ganda

No	Citra Pengujian	Hasil Identifikasi	
		Berhasil	Gagal
1		4 tabel	0
2		4 tabel	0
3		4 tabel	0
4		4 tabel	0
5		2 tabel	2 tabel
6		4 tabel	0
7		4 tabel	0

8		4 tabel	0
9		4 tabel	0
10		4 tabel	0
11		4 tabel	0
Total Tabel pengujian		44	
Identifikasi Berhasil		42	
Identifikasi Gagal		2	

Data tabel pengujian di atas didapatkan nilai variabel pengujian sebagai berikut :

TP = 42; FP = 2; TN = 0 ; FN = 0

Dengan variabel di atas didapatkan nilai akurasi dengan melakukan perhitungan menggunakan rumus *confusion matrix* di bawah ini :

- Akurasi = $((TP + TN) / (P + N)) \times 100$
- Akurasi = $(42 / 44) \times 100$
- Akurasi = $(0.9545) \times 100$
- **Akurasi = 95.45 %**

Pengujian jarak penggunaan program

Pengujian jarak ini perlu dilakukan untuk mengetahui jarak maksimum dan minimum yang diperlukan agar program dapat berjalan dengan semestinya tanpa terjadi masalah seperti tidak mengenali lembar jawaban ataupun tabel pilihan ganda. Dikarenakan keterbatasan alat pengujian, pengujian terhadap pengaruh intensitas cahaya tidak dapat dilakukan, karenanya pada pengujian jarak ini dilakukan pengujian dengan informasi pencahayaan sebagai berikut :

Lampu LED dengan spesifikasi :

- Tegangan : AC 80 – 260 V, 50-60 Hz
- Daya : 22 W
- Index pencahayaan : > 70 Ra

Selain itu pengujian juga dibantu dengan alat bantu berupa tripod dan kamera external berupa *smartphone*, dengan sudut kamera terhadap objek sebesar 90 derajat. Parameter pengujian ini dilakukan dengan cara memposisikan tabel jawaban pilihan ganda dari beberapa dataset pengujian di hadapan kamera yang sudah diatur ketinggiannya. Tingkat keberhasilan pengujian dihitung dengan menggunakan rumus berikut:

$$\text{Tingkat keberhasilan} = (X / Y) \times 100 \quad (3)$$

Keterangan :

- X = total dataset yang jawabannya berhasil diidentifikasi
- Y = total dataset yang diuji

Berikut adalah tabel hasil dari pengujian jarak untuk jarak 17 cm hingga jarak 27 cm.

Tabel 4. Hasil pengujian jarak program

No	Jarak	Jumlah pengujian	Tingkat keberhasilan
1	17 cm	10 dataset	100%
2	18 cm	10 dataset	100%
3	19 cm	10 dataset	100%
4	20 cm	10 dataset	100%
5	21 cm	10 dataset	100%
6	22 cm	10 dataset	100%
7	23 cm	10 dataset	100%
8	24 cm	10 dataset	100%
9	25 cm	10 dataset	90%
10	26 cm	10 dataset	70%
11	27 cm	10 dataset	0%

Dari hasil tingkat keberhasilan tabel pengujian di atas, dapat disimpulkan bahwa jarak ideal untuk menggunakan program adalah kisaran 17 cm hingga 24 cm. Program tidak mampu dalam melakukan pengidentifikasian jawaban dalam tabel jawaban pilihan ganda semata-mata karena tidak sesuainya jarak dengan sistem yang dirancang. Jika program tidak mampu untuk mendeteksi tabel jawaban salah satu kemungkinan penyebabnya adalah *contour* yang tidak tersambung (terputus) sehingga menyebabkan tabel pilihan ganda tidak dapat diidentifikasi, sebagaimana menurut Yusril Ihsan 10 Hal lain juga dapat disebabkan karena sistem tidak dapat mendeteksi ukuran batas dari pinggir lembar jawaban karena jarak yang terlalu jauh.

Pengujian keakuratan program secara realtime

Pengujian ini dilakukan dengan cara dataset lembar jawaban untuk pengujian akan dihadapkan di depan kamera eksternal dengan bantuan alat bantu tripod, dengan sudut kamera terhadap lembar jawaban sebesar 90 derajat dengan menerapkan jarak ideal (sekitar 17 cm) dan tingkat pencahayaan yang sudah disebutkan pada pengujian

sebelumnya, hasil dari pengecekan dengan program berupa keberhasilan program dalam mendeteksi jawaban yang ditandai oleh siswa sebagai jawaban yang terpilih. Penilaian akurasi masih akan dihitung dengan menggunakan persamaan rumus *confussion matrix*, dimana untuk variabel nya dapat dilihat pada poin-poin berikut ini.

- TP = (*True Positive*) Saat jawaban yang dipilih oleh siswa diidentifikasi sebagai jawaban yang terpilih
- FP = (*False Positive*) Saat jawaban yang dipilih oleh siswa tidak diidentifikasi sebagai jawaban yang terpilih
- TN = (*True Negative*) Saat jawaban yang tidak dipilih oleh siswa tidak diidentifikasi oleh program
- FN = (*False Negative*) Saat jawaban yang tidak dipilih oleh siswa teridentifikasi sebagai jawaban yang terpilih

Hasil pengujian per dataset lembar jawaban tidak akan ditampilkan dalam tabel karena jumlahnya yang banyak, melainkan tabel kesimpulan yang akan disajikan, sebagaimana tabel kesimpulan pengecekan berikut ini.

Tabel 5. Hasil pengujian tingkat keakuratan program

Jumlah Dataset	Total Tabel	Total Jawaban	Identifikasi Jawaban	
			Berhasil	Gagal
40 dataset	160	2000	1877	123

Dari tabel di atas didapatkan variabel berikut :

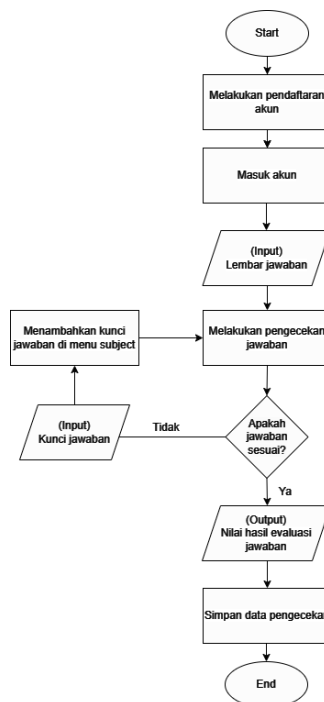
- TP = 1877
- TN = 123
- $(TP / (P + N)) \times 100$

Maka didapatkan akurasi dari pengujian ini adalah sebesar : **93.85 %**

Pengujian program untuk dapat dengan mudah dimengerti

Pada pengujian ini, selain menguji tampilan hasil deteksi, seluruh antarmuka aplikasi juga akan diuji untuk memastikan aplikasi ini dapat digunakan dengan baik oleh berbagai pengguna tanpa mengalami masalah.

Adapun untuk sistematis alur penggunaan aplikasi dapat dilihat pada gambar berikut :



Gambar 13. Alur penggunaan aplikasi

Penjelasan dari gambar alur penggunaan aplikasi di atas dapat dilihat pada poin-poin berikut ini :

- Mula-mula pengguna diminta untuk membuat akun, akun ini berfungsi sebagai penyimpan pengaturan dan data kunci jawaban,
- Setelah membuat akun, pengguna masuk ke dalam aplikasi dengan akun yang baru saja dibuat.
- Saat pertama kali dipakai, aplikasi tidak memiliki data apapun, karena itu pengguna diminta untuk memasukkan data kunci jawaban.
- Setelah data kunci jawaban ditambahkan, pengguna dapat memulai pengecekan jawaban kapan saja, dengan menggunakan kamera webcam maupun eksternal.
- Pengguna juga dapat memilih apakah data hasil pengecekan akan disimpan atau tidak.
- Saat pengguna menutup aplikasi, dan membukanya kembali suatu saat, aplikasi akan memeriksa apakah pengguna sebelumnya sudah keluar dari akun belum, jika belum maka aplikasi akan langsung membuka menu utama.

Untuk file hasil penyimpanan dari hasil pengecekan digunakan file berekstensi .xls yang cukup familiar di kalangan guru, sehingga dalam penggunaan aplikasi ini tidak banyak keperluan untuk menginstal aplikasi tambahan.

Pada pengujian ini, responden akan diminta untuk mengisi kuesioner sebagai bagian dari evaluasi. Hasil pengisian kuesioner akan diolah menggunakan metode SUS (*System Usability Scale*) yang membutuhkan teknik khusus dalam penyusunan kuesioner. Berikut adalah isi kuesioner yang disusun dan didistribusikan kepada responden.

Tabel 6. Isi pertanyaan kuesioner

No	Pertanyaan
1	Saya berpikir akan menggunakan sistem ini lagi
2	Saya merasa sistem ini rumit untuk digunakan
3	Saya merasa sistem ini mudah digunakan
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini
5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya
6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini)
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat
8	Saya merasa sistem ini membingungkan
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini

Aturan-aturan yang terdapat dalam menggunakan metode SUS diantaranya adalah :

- Setiap pertanyaan ganjil, skor yang didapat setiap nomor dikurangi dengan nilai 1
- Setiap pertanyaan genap, nilai 5 dikurangi dengan skor yang didapat
- Total skor hasil perhitungan sebelumnya, dijumlah dan dikali dengan 2,5

Dengan aturan perhitungan diatas maka didapatkan hasil perhitungan SUS sebagai berikut.

Tabel 7. Skor asli hasil tanggapan responden

NILAI	Nomor Pertanyaan									
	1	2	3	4	5	6	7	8	9	10
	5	2	5	1	4	1	4	2	3	1
	5	2	5	3	4	2	5	1	4	2
	5	2	3	3	4	1	5	2	4	2
	5	1	4	1	4	2	5	1	4	1
	4	2	5	1	4	1	5	2	4	1
	5	2	5	1	4	1	5	2	4	1

Tabel skor asli hasil tanggapan responden sebelumnya akan dihitung dengan perhitungan SUS (*System Usability Scale*). Perhitungan SUS digunakan untuk menganalisis tingkat kegunaan dan kemudahan penggunaan sistem yang telah diuji oleh responden sebelumnya. Metode SUS memberikan cara objektif untuk menilai kesan keseluruhan pengguna terhadap sistem dengan mempertimbangkan aspek-aspek seperti efisiensi, pembelajaran, dan kepuasan pengguna. Dengan demikian, didapatkan wawasan yang berharga tentang bagaimana pengguna sebelumnya merespons dan berinteraksi dengan sistem yang telah diuji. Berikut adalah hasil dari perhitungan SUS dari hasil tanggapan responden.

Tabel 8. Skor setelah diolah dengan metode SUS

Nomor Pertanyaan										total	Total * 2.5
1	2	3	4	5	6	7	8	9	10		
4	3	4	4	3	4	3	3	2	4	33	82,5
4	3	4	2	3	3	4	4	3	3	31	77,5
4	3	2	2	3	4	4	3	3	3	36	90
4	4	3	4	3	3	4	4	3	4	35	87,5
3	3	4	4	3	4	4	3	3	4	36	90
4	3	4	4	3	4	4	3	3	4	34	85
Jumlah / banyak tanggapan										85,41666667	

Berdasarkan nilai dari tabel skor setelah diolah dengan menggunakan metode SUS, didapatkan hasil nilai akhir adalah 85,4166667 yang menandakan bahwa aplikasi termasuk kategori yang lulus pengujian kelayakan, mudah digunakan dan tidak membutuhkan waktu lama untuk menyesuaikan diri.

4. Kesimpulan

Sistem pengecekan lembar jawaban yang tidak berbasis komputer dikembangkan dengan bahasa pemrograman python dan library *OpenCV*. Berbagai tahapan pengolahan data, termasuk *thresholding*, *edge detection*, *contour sorting*, dan *warp perspective*, digunakan untuk mencapai hasil pengecekan jawaban pada lembar jawab yang optimal. Sejumlah teknik pengujian telah dilakukan, dari pengujian kesesuaian program dengan posisi tabel jawaban hingga kemampuan program dalam menampilkan hasil pengecekan dengan jelas dan mudah dipahami. Meskipun akurasi tidak bisa dijamin 100%, program ini memberikan hasil yang cukup akurat dan dapat membantu mempercepat proses pengecekan jawaban dengan efisien. Dalam pengujian keakuratan menggunakan metode *contour sorting*, akurasi mencapai 93% dari 40 dataset pengujian. Selain itu, menggunakan 11 dataset baru untuk menguji posisi tabel jawaban pilihan ganda, tingkat akurasi posisi mencapai 95%, dan berdasarkan pengujian jarak ideal, diketahui bahwa jarak ideal kamera ke lembar jawaban adalah dalam rentang 17cm s.d. 24cm.

Referensi

1. Purwanto, *Evaluasi Hasil Belajar*. Pustaka Pelajar, 2011.
2. M. Syah, *Psikologi Pendidikan dengan Pendekatan Baru*. Bandung: PT Remaja Rosdakarya, 2010.
3. N. Purwanto, *Ilmu pendidikan teoritis dan praktis*. Bandung: Remaja Karya, 2022.
4. S. van der Walt *et al.*, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, Jun. 2014.
5. P. Boby, "Paper Real Time System," 2018.
6. A. E. Tümer and Z. Küçükara, "An Image Processing Oriented Optical Mark Recognition and Evaluation System," *Int. J. Appl. Math. Electron. Comput.*, vol. 6, no. 4, pp. 59–64, 2018.
7. S. H. Sdgd, N. Dqj, and E. Flwud, "Phqjxqnddq \$ Ojrulwpd & Dqq \ ' Hwhfwlrq," pp. 5–10.
8. M. N. Hermawan, "Deteksi Lembar Jawaban Komputer Menggunakan Omr (Optical Mark Recognition) Di Mts Nurul Iman," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 3, pp. 1361–1372, 2021.
9. G. Qorik *et al.*, "Deteksi Pilihan Jawaban Otomatis pada Lembar Jawaban Komputer Menggunakan Metode Image Thresholding dan Contour Shorting," *COREAI J. Kecedasan Buatan, Komputasi dan Teknol. Inf.*, vol. 1, no. 1, pp. 1–9, 2020.
10. Y. I. Hernafi, T. A. Riza, and H. Hafidudin, "Aplikasi Android Koreksi Lembar Jawaban Komputer Menggunakan Opencv," *eProceedings Appl. Sci.*, vol. 6, no. 1, pp. 447–489, 2020.
11. M. Al Faruqi, "Sistem Pemetaan Posisi Objek Kendaraan Menggunakan Pengolahan Citra Pada Area 360°," 2021.
12. M. C. Swedia, Ericks Rachmat, "Algoritma Transformasi Ruang Warna," *Univ Gunadarma*, 2010.
13. M. R. V. Aditya, N. L. Husni, D. A. Pratama, and A. S. Handayani, "Penerapan Sistem Pengolahan Citra Digital Pendeteksi Warna pada Starbot," *J. Tek.*, vol. 14, no. 02, pp. 185–191, 2020.
14. H. Mulyawan, M. Z. H. Samsono, and Setiawardhana, "Identifikasi Dan Tracking Objek Berbasis Image," pp. 1–5, 2011.
15. Sylvain Paris, Pierre Kornprobst, "*Bilateral Filtering: Theory and Applications*," *Computer Graphics and Vision*, vol. 4, no. 01 1–73, 2008.